POCKET COMPUTER NEWSLETTER

(c) Copyright 1983 POCKET COMPUTER NEWSLETTER

Issue 27 - September

RADIO SHACK ANNOUNCES THE PC-3

Tandy Corporation has announced the availability of a 4-ounce shirt-pocket computer that is programmable in BASIC.

The new PC has a 24-character liquid crystal display, a display accuracy up 10 digits and 1.4K of user-programmable memory. Besides operating in a programmable mode, it can be used as a direct entry calculator.

The PC-3 is said to be software compatible with the earlier PC-1. Thus, Radio Shack claims that their existing library of programs produced for the PC-1 will be immediately available for use on it.

The U.S. price of the PC-3 is set at $99.95 from Radio Shack computer centers and stores. It is supplied with batteries and an instruction manual. *(Note: the PC-3 appears to be a custom-labeled version of the Sharp PC-1250.)*

A matching accessory, termed the PC-3 Printer/Cassette Interface is reported to be available at a U.S. price of 119.95. It enables the PC-3 pocket computer to load and store data using a cassette tape recorder. A thermal dot-matrix printer capable of printing 24 characters to a line at a rate of one line per second, provides for hardcopy of programs and data. The unit comes supplied with rechargeable batteries, an A.C. adaptor, some paper, cables to connect to a recorder unit and an operating manual.

Another *Personal Information*(TM) product.

PAPER HOLDER KIT FOR PC-1500 & PC-2

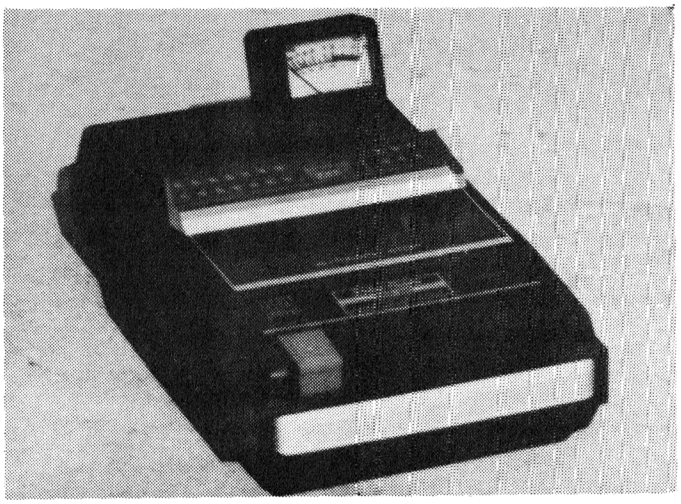
The *Bisi Beaver Paper Holder Kit* is an attachment for Radio Shack PC-2 and Sharp PC-1500 computers that will accomodate a standard 2-1/4 inch roll of calculator paper. The kit includes the paper holder, a template for proper alignment and the drilling of holes in the printer cover, and instructions for making the modifications. When not in use the paper holder and paper may be stored in the computer's carrying case.

The kit retails in the U.S. for $24.95 (plus $1.56 sales tax in the state of Washington). Kits are normally shipped by UPS with shipping charges collect. Orders may be placed with: *Bisi Beaver Software Company, P.O. Box 53, Vashon, WA 98070.*

CASSETTE VOLUME MONITOR

If you use a standard audio cassette recorder to store computer programs, you know it can sometimes be difficult to get the volume adjusted properly when loading programs.

A new electronic device named *VU-LOAD* may be what you need to eliminate such problems. It enables you to "observe" the signal level at the ear jack of the cassette recorder. You may then adjust the level for the optimum volume setting.



If your cassette does not have an automatic level control, it is claimed that the *VU-LOAD* monitor will enable you to find the optimum volume setting at which your cassette recorder will "save" your programs. And, it will give a positive "save" indication with most other cassette players.

VU-LOAD may be mounted anywhere within reach of its 16 inch connector, which plugs into the ear jack of a cassette recorder.

The U.S. price is quoted as $20.95 plus $2.50 for postage/handling. It is claimed to be fully guaranteed and to be available from: *L & G Enterprises, Box 6854, Silver Spring, MD 20906.*

VARIABLES LISTER

This is a cross reference program for the Radio Shack PC-2 and Sharp PC-1500. It will list all the variables used in a program, showing them in alphabetical order (ascending ASCII). After each variable name is listed, the line(s) where that variable is (are) used will be shown. If a variable is used several times within a line (such as in A=A+1) then the line number is repeated the appropriate number of times.

The program distinguishes between all four possible variations of a variable assignment (i.e., A, A(), A$() and A$).

A *linked list* was used to conserve memory. Three major arrays are used by the program. T$ holds the variables used in the cross referenced program. M stores the starting node in the linked list for each variable name. O stores the linked list.

T$(n) = **Name of variable n.**

M(n) = **Index in array O of the first entry for the line numbers associated with variable n.**

O(1,M(n)) = **First line number associated with variable n.**

O(2,M(n)) = **Index in O of the next reference in O associated with variable n. That is, the next line number associated with variable n will be stored at O(1,O(2,M(n))). O(2,O(M(n))) holds the index in O of the next reference to the variable n. A value of zero in O(2,M(m) indicates the last reference to that variable.**

Consider the short example program here:

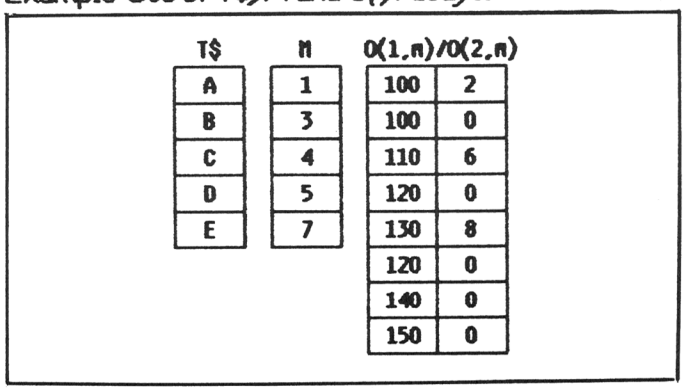
```
 90 : REM TEST
100 : A = A + 1
```

```
110 : B = 12
120 : C = B + 13
130 : D = 14
140 : E = 15
150 : D = 0
160 : END
```

The accompanying diagram illustrates how the T$, M and O$ arrays in the Variables Lister program would appear when processing the above example program.

The arrays T$ and M have been dimensioned to 25 and the linked list to 250. This currently limits

Example *Use of T$, M and O() Arrays.*



T$	M	O(1,n)	O(2,n)
A	1	100	2
B	3	100	0
C	4	110	6
D	5	120	0
E	7	130	8
		120	0
		140	0
		150	0

Program *Variables Lister.*

```
1000:REM  "XREF"
2010:CLEAR :T=1:N
     =1:WAIT 10
2020:DIM M(25),O(
     2,250),T$(25
     )*4
2030:DIM N$(0)*36
2040:FOR I=1TO 25
     :M(I)=0:T$(I
     )="":NEXT I
2050:FOR I=1TO 25
     0:O(1,I)=0:O
     (2,I)=0:NEXT
     I
2080:Q=PEEK &7869
     *256+PEEK &7
     86A
2120:Q=Q+5
2140:REM
2160:N$(0)=N$(0)+
     CHR$ (PEEK Q
     )
2170:Q=Q+1
2190:IF PEEK Q<>&
     0DGOTO 2140
2220:LPRINT N$(0)
     :LF 1:Q=Q+1:
     W$=""
2250:REM
2260:IF PEEK Q=&F
     FGOTO 2610
2280:L=PEEK Q*256
     +PEEK (Q+1):
     Q=Q+3:PRINT
     L
2290:IF PEEK Q=&F
     1AND PEEK (Q
     +1)=&ABTHEN
     LET Q=Q+PEEK
     (Q-1)-1
2310:REM
2320:IF PEEK Q=&0
     DGOTO 2560
2350:IF PEEK Q>&E
     5THEN LET Q=
     Q+2:GOTO 254
     0
2380:IF PEEK Q<&4
     1OR PEEK Q>&
     90GOSUB 6000
     :Q=Q+1:GOTO
     2540
2410:REM
2430:W$=W$+CHR$
     PEEK Q:Q=Q+1
2450:IF (&41<PEEK
     QAND PEEK Q<
     &90)OR PEEK
     Q=&24GOTO 24
     10
2480:IF PEEK Q=&2
     8THEN LET W$
     =W$+CHR$
     PEEK Q:Q=Q+1
2510:P=1:GOSUB 30
     00:W$=""
2540:REM
2550:GOTO 2310
2560:REM
2580:Q=Q+1
2600:GOTO 2250
2610:REM
2630:T=T-1:LPRINT
     "WORDS FOUND
     :";T
2640:N=N-1:LPRINT
     "REFERENCES:
     ";N
2650:GOSUB 4000:
     GOSUB 5000
2670:END
3000:REM "STOTRE"
3020:IF T>1GOTO 3
     080
3040:T$(T)=W$:M(T
     )=N:O(1,N)=L
3050:T=T+1:N=N+1
3070:GOTO 3400
3080:REM
3090:REM
3100:IF (P>T)OR (
     W$=T$(P))
     GOTO 3150
3120:P=P+1
3140:GOTO 3090
3150:REM
3170:IF P>TGOTO 3
     340
3200:P=M(P)
3220:REM
3230:IF O(2,P)=0
     GOTO 3280
3250:P=O(2,P)
3270:GOTO 3220
3280:REM
3300:O(2,P)=N:O(1
     ,N)=L:N=N+1
3330:GOTO 3380
3340:REM
3350:T$(T)=W$:M(T
     )=N:O(1,N)=L
3360:T=T+1:N=N+1
3380:REM
3400:REM
3410:RETURN
4000:REM "SORT"
4020:IF T<=1GOTO
     4230
4050:FOR I=1TO T-
     1
4070:K=I:C=I+1
4100:FOR J=CTO T:
     IF T$(K)>T$(
     J)THEN LET K
     =J
4110:NEXT J
4140:IF I=KGOTO 4
     190
4160:U$=T$(I):U=M
     (I)
4170:T$(I)=T$(K):
     M(I)=M(K):T$
     (K)=U$:M(K)=
     U
4190:REM
4200:NEXT I
4230:REM
4240:RETURN
5000:REM "PRTLIN"
5020:FOR I=1TO T
5040:P=M(I):J=1:
     LPRINT T$(I)
5050:REM GOSUB 70
     00
5070:REM
5090:LPRINT USING
     "######";O(1
     ,P);:P=O(2,P
     ):J=J+1
5110:IF P<>0GOTO
     5070
5130:LPRINT
5140:NEXT I
5160:RETURN
6000:REM "SP CHR"
6020:IF PEEK Q<>&
     22GOTO 6110
6040:REM
6060:Q=Q+1
6080:IF PEEK Q<>&
     22GOTO 6040
6110:REM
6130:IF PEEK Q<>&
     26GOTO 6250
6150:REM
6170:Q=Q+1
6190:IF (PEEK Q>=
     &30AND PEEK
     Q<=&39)OR (
     PEEK Q>=&41
     AND PEEK Q<=
     &46)GOTO 615
     0
6220:Q=Q-1
6250:REM
6260:RETURN
STATUS 1        1542
```

Example *RUN of Variables Lister.*

```
          TEST

WORDS FOUND: 5
REFERENCES: 8
A
   100    100
B
   110    120
C
   120
D
   130    150
E
   140
```

the program to processing 25 different variables with no more than 250 total references. If these dimensions are not satisfactory, they may be altered. Remember, however, that increasing the size of these arrays will decrease the amount of memory left available for an application program. To change the dimension of T$ and M, change lines 2020 and 2040. Lines 2020 and 2050 are modified to change the dimension of O().

The program assumes that the first line of the application program begins with a REM statement. The contents of this REM are printed as a heading on the variables listing. It is a good idea to place the name of the program in this line.

The LCD contains the line number being processed during operation of the program.

Operation is straightforward:

1. Load XREF into memory.
2. MERGE your program.
3. RUN.

Thanks for this valuable utility program go to: *Diane Campbell, 220 Houston, League City, TX 77573.*

VERTICAL LISTER

The SIDELISTER program published in Issue 22 of *PCN* introduced the concept of vertical printing of BASIC program lines. As *Mel Beckman*, author of the article pointed out, his pioneering effort was slow in operation.

This program uses machine language to expand tokens into a string of characters. It will provide a vertical listing at about the same speed as a regular LLIST command.

Making the Vertical Lister Cassette

First enter the machine codes shown in the accompanying listing using either POKEs or a monitor program. (As with any machine language program, the correct entry of these codes is absolutely crucial. Check your work.) Save the ML program on cassette by executing:

CSAVE M "VERT LISTER (ML)";&7650,&76EE

Next, clear memory with the NEW command. Now enter the BASIC portion of the Vertical Lister as shown in the accompanying listing. Add this to the same cassette using the standard procedure:

CSAVE "VERTICAL LISTER"

Program *ML Portion of Vertical Lister.*

```
7650 B5 2A ED 79
7654 F4 01 8B 02
7658 B5 50 AE 77
765C 4F FD 88 CC
7660 99 B5 07 FD
7664 CA FD 5A 04
7668 AE 77 4E 6A
766C FB CD BA FD
7670 0A 45 28 45
7674 2A FD 88 CD
7678 10 40 FD 0A
767C B5 3A 51 44
7680 45 B7 E0 83
7684 0D B7 0D 8B
7688 07 FD 88 46
768C B5 01 8E 24
7690 FB 9A 28 45
7694 2A FD 88 48
7698 B0 4A 54 45
769C B9 0F 8B 44
76A0 FD CA 45 A6
76A4 89 39 05 26
76A8 89 35 46 46
76AC 46 05 D9 91
76B0 05 45 B9 0F
76B4 2A DF 28 A5
76B8 77 4E F9 A3
76BC 77 4F 16 91
76C0 07 20 16 89
76C4 02 68 00 83
76C8 08 22 1A B5
76CC 20 51 51 51
76D0 51 62 F5 88
76D4 03 A4 8B 03
76D8 B5 20 51 FD
76DC 0A 9E 5F 44
76E0 44 44 9E 49
76E4 84 B3 08 08
76E8 B7 C8 91 53
76EC FD 1A E4 00
```

Next, clear memory with the NEW command. Now enter the BASIC portion of the Vertical Lister as shown in the accompanying listing. Add this to the same cassette using the standard procedure:

CSAVE "VERTICAL LISTER"

Using the Vertical Lister

Load the BASIC program you desire to have listed into the computer. Make sure that STATUS 0 yields a value of at least 455. That is the amount of space required by the Vertical Lister program, including the space it uses for the dimensioned variable A$().

Program *BASIC Portion of Vertical Lister.*

```
1:"VL"INPUT "SIZ
  E? ";S:GRAPH :
  CSIZE S:ROTATE
  1:A=STATUS 2-
  STATUS 1:DIM A
  $(3*S-1)*76/S+
  4:X=216
2:CALL &7650,A:C
  =0
3:IF X=0GLCURSOR
  (0,-576):SORGN
  :X=216
4:X=X-9*S:
  GLCURSOR (X,0)
  :LPRINT A$(C):
  C=C+1:IF C<3*S
  IF A$(C)GOTO 3
5:IF PEEK A<255
  GOTO 2
6:GLCURSOR (0,-5
  76):TEXT
STATUS 1
                196
```

Insert the cassette containing the two parts of the Vertical Lister program into the tape player. Execute CLOAD M. After the machine language portion as loaded, execute MERGE to bring in the BASIC part of Vertical Lister. Now execute the command RUN "VL" and enter the desired print size (1 or 2) in response to the prompt. Your BASIC program will then be listed in vertical format. (Of course, the Vertical Lister program itself is not listed.)

Some Alternatives

The procedure described above makes it easy to produced vertical listings of programs that you have previously saved on tape. The Vertical Lister program itself takes only a small amount of memory and thus takes less than a minute to load. Furthermore, locating the ML portion in the strings variable area means it is unnecessary to have previously reserved memory space for it.

However, when used in the above manner, the VL program can not be used as an aid in debugging a program that is in the process of being developed. This is because the MERGE of the BASIC portion of the VL program prevents editing of the lines of the BASIC program that is under development. Also, string variables E$ to N$ must be left alone, since the area they reside in is occupied by the ML portion of Vertical Lister. (Note that the use of CLEAR would completely erase the ML portion of VL, too!)

One alternative to this situation is to locate the ML portion of the program elsewhere. Indeed, the ML routine has been written so that it is relocatable without any modification to it. (Of course, the CALL &7650 statement in line 2 of the BASIC part would have to be changed to conform to the new address of the ML routine.) If Reserve memory is not being used for anything else, it would be a good place for the ML routine. Another possibility would be to relocate the beginning of the BASIC storage area using the NEW XXXX statement. (The ML code for VL requires 159 bytes.)

If you desire to avoid the use of a MERGE in order to leave the BASIC program in a form whereby it can be edited, then the six lines of BASIC programming used by VL may be loaded prior to development of the program. (As an alternate, the BASIC portion of VL can manually be typed in at any appropriate point.) Naturally, when a MERGE command is not used to load the BASIC portion of VL, then Vertical Lister itself *will* be included in the vertical listing.

A closing note: This program only searches the token tables of the PC-1500 and CE-150. It will not look for tokens in the CE-158 or other devices. Hence BASIC programs that use statements unique to such peripherals can not be listed by VL.

This program developed by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

THREE-DIMENSIONAL PROJECTIONS

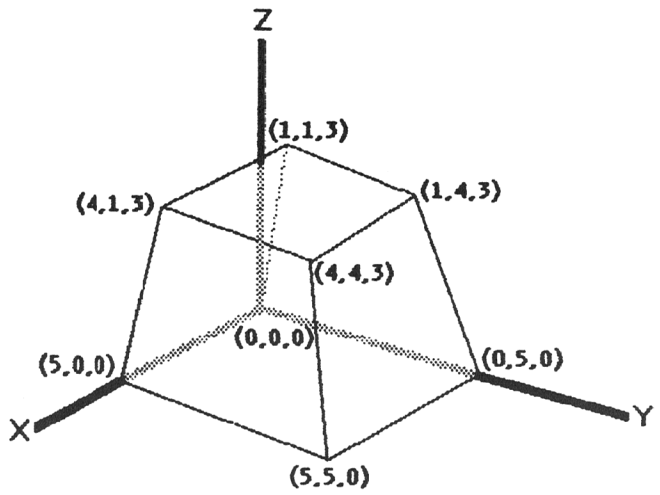
This program uses the PC-1500/PC-2 with its printer to plot the projection of a three-dimensional object onto a plane. Two kinds of projections are available. One is an *oblique* projection, in which the plane of projection is parallel to the XY plane. In the second type, an *axonometric* projection, the projecting rays are perpendicular to the plane of projection.

In either type of projection, the user selects the direction of the projecting rays by inputting values of L, M and N. These three values represent, respectively, the X-, Y- and Z-components of the direction that the projecting lines are to have. You may think of (L,M,N) as a vector in the direction of the projecting lines.

Taking the points in the order in which they are to be connected, the X-, Y- and Z-coordinates of each point must be specified in DATA statements. The letter S should appear in the DATA list to indicate the *start* of each sequence of points to be connected. (Its effect should become clear when the accompanying example is studied.) Use the letter T as the last item in the DATA list to signal *termination* of the drawing.

An Example

The truncated pyramid in the accompanying diagram is shown with the X-, Y- and Z-coordinates labeled at each of the points to be connected. These coordinates appear in the DATA statements in lines 100 to 103 of the program listing. Note, also, the use of S and T in these program lines. The letter S always precedes a sequence of points that are to be joined by line segments.



When the program is RUN, it first requests the input of the letter A or O. This indicates whether an *axonometric* or *oblique* projection is to be drawn. Next, the direction of the projecting lines must be specified by L, M and N. Note that L, M and N should be positive numbers. For an oblique projection, the use of 7, 4 and 3 results in a drawing with a fairly pleasing appearance. With axonometric projections, the choice 4, 3, 2 works well. (The "ideal" direction will depend on the object being represented.)

Finally, the program requests maximum and minimum values of X, Y and Z. These values determine the portion of three-dimensional space that is to be represented in the projection drawn. They are not necessarily the largest and smallest values of a coordinate used in the DATA statements. To retain correct scaling, the same size interval should be used in all three dimensions. To illustrate: the minimum values of X, Y and Z might all be zero, with the maximum values stated as 20, even if all three do not attain that value. (Of course, there may be cases where it is desirable *not* to use the same scale in all three directions. Perhaps a magnification is wanted in one dimension, for example.)

If you desire to re-execute the program, it is not necessary to re-enter those values that are not being altered. For instance, if you want to change just the direction of view, simply key ENTER without data when the prompts appear for the type of projection and the maximum and minimum values.

After the drawing has been completed, you have the option of keying ENTER to print out a listing of the parameters used in the drawing.

Program *Three-Dimensional Projections.*

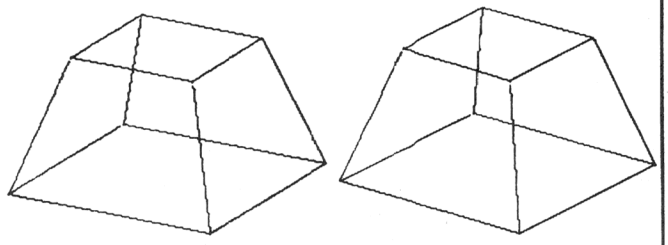
```
10:INPUT "Type of
   proj (A/O)? "
   ;A$
11:INPUT "Directi
   on: L? ";L,"M?
    ";M,"N? ";N
12:INPUT "Min X?
   ";A,"Max X? ";
   B
13:INPUT "Min Y?
   ";C,"Max Y? ";
   D
14:INPUT "Min Z?
   ";E,"Max Z? ";
   F
20:CLS :IF A$="A"
   GOTO 30
21:K1=216/(L+M):K
   4=0
22:XH=K1*M/(B-A):
   YH=K1*L/(D-C):
   XV=K1*N/(B-A):
   ZV=K1*L/(F-E):
   YV=0:GOTO 40
30:K1=L*L+M*M:K2=
   216/√(2*K1):K3
   =K2/√(K1+N*N):
   K4=108-K2*(L+M
   )/2
31:XH=K2*M/(B-A):
   YH=K2*L/(D-C):
   XV=K3*L*N/(B-A
   ):YV=K3*M*N/(D
   -C):ZV=K1*K3/(
   F-E)
40:S=1E99:T=2E99:
   GRAPH :
   GLCURSOR (K4,-
   270):SORGN
41:READ X:IF X=T
   GOTO 44
42:IF X<SREAD Y,Z
   :GOSUB 60:LINE
   -(HC,VC):GOTO
   41
43:READ X,Y,Z:
   GOSUB 60:
   GLCURSOR (HC,V
   C):GOTO 41
44:GLCURSOR (0,-8
   0):TEXT
50:PRINT "Key ENT
   ER to list inp
   uts"
52:LPRINT "L=";L:
   LPRINT "M=";M:
   LPRINT "N=";N:
   LPRINT
53:LPRINT "Min X=
   ";A:LPRINT "Ma
   x X=";B:LPRINT
   "Min Y=";C:
   LPRINT "Max Y=
   ";D:LPRINT "Mi
   n Z=";E:LPRINT
   "Max Z=";F
54:END
60:HC=XH*(B-X)+YH
   *(Y-C):VC=XV*(
   B-X)+YV*(D-Y)+
   ZV*(Z-E):
   RETURN
100:DATA S,0,0,0,5
    ,0,0,5,5,0,0,5
    ,0
101:DATA 0,0,0,1,1
    ,3,4,1,3,4,4,3
102:DATA 1,4,3,1,1
    ,3,S,4,1,3,5,0
    ,0
103:DATA S,4,4,3,5
    ,5,0,S,1,4,3,0
    ,5,0,T
STATUS 1      878
```

Special Cases

An oblique projection having a foreshortening factor of 1/2 is called a *cabinet* drawing. An example of a cabinet drawing may be produced by using L = 10, M = 4 and N = 3 in the program.

An *isometric* projection is an axonometric

Example *RUNs of 3-Dimensional Projections.*



```
L= 7                    L= 4
M= 4                    M= 3
N= 3                    N= 2

Min X= 0                Min X= 0
Max X= 5                Max X= 5
Min Y= 0                Min Y= 0
Max Y= 5                Max Y= 5
Min Z= 0                Min Z= 0
Max Z= 5                Max Z= 5
```

projection in which the projecting lines make equal angles with the X-, Y- and Z-axes. This kind of projection results when L, M and N are equal.

This program provided by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

COUNT-DOWN TIMER PROGRAM

OK, all you PC-1/PC-1211 fans, here is a little routine that will enable you to use your PC as a timer that beeps when a specified interval has elapsed:

```
10  BEEP 1:PAUSE " TIMER PROGRAM"
20  Y=12099.24205
30  INPUT "TIME (H.MMSS):";T:W=INT(Y*DEG
    T + 80000 - 2)
40  W=W-1:IF W>80000 GOTO 40
50  BEEP 3:PAUSE "TIME IS UP!":GOTO 50
100 "C":PAUSE "CALIBRATION -->":USING
110 INPUT "TRUE TIME(H.MMSS):";S
120 X=DEG T/DEG S:X=X*Y
130 PRINT "NEW Y= ";X:END
```

Example Run

Want to know when exactly 14 minutes and 35 seconds have elapsed? Start the program with a RUN command:

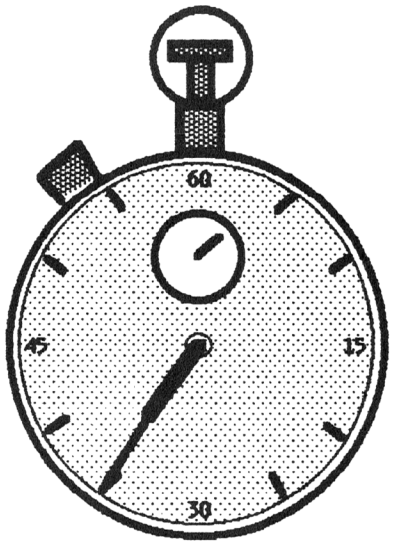
```
RUN
TIMER PROGRAM
TIME (H.MMSS): 0.1435
. . . . 14 minutes, 35 seconds later . . . .
Beep . . . Beep . . . Beep
TIME IS UP!
```

Calibration

If, after a trial run of a few hours, you find that the timer is off, you may calibrate the program against another clock or watch. Say that on a test run of exactly 3 hours, you find that the program starts to beep after a true time of 2 hours, 59 minutes and 37 seconds. Just run the calibration program as shown here:

```
RUN 100   (or SHIFT/C in the DEF mode)
CALIBRATION -->
TRUE TIME (H.MMSS): 2.5937
NEW Y= 12125.06395
```

After executing the calibration routine, you would need to place your PC into the PRO mode and change the calibration constant Y in line 20 to the new value. Your PC will now be custom-calibrated!



Sharp PC-1250/1251 and Radio Shack PC-3 owners: setting Y = 46141.17733 in line 20 seems to yield pretty good results for this program when installed in one of these models. You should be able to tweek the timing up, if necessary, using the calibration subroutine. Try it!

You can use your portable timer to do a three-minute egg, remind you when your parking meter is about to expire or when it is time to gracefully exit from the speaker's podium!

This program was skillfully crafted by: *Emerich Auersbacher, 41 King Street #2, Belleville, NJ 07109.*

FROM THE WATCH POCKET

Where are the 16K RAMs? On the way, is the latest word, and expected to be available in the U.S. by mid-September. The list price is still quoted as $169.95. Be careful if you plan to upgrade to a 16K module and are using machine language routines. It may be that the 16K module is addressed such that it would not be compatible with your present address-dependent machine language programs! If you use strictly BASIC programs, however, you should be able to simply plug in the new 16K module and load and execute programs with nothing to worry about except how to use up all that lovely RAM!

Of course, the introduction of the new 16K modules is likely to put the heat on anybody thinking about producing some kind of better program/data loading device. Filling up all that memory using an audio tape recorder can take 20 - 30 minutes. That can begin to get a bit aggravating for those who are trying to use PCs to save time! Where is a nice wafer drive or *anything* that will quickly load stuff into a PC-2/PC-1500?

Interested in trying to make money with your computer? **Computer Business Information Exchange**, *P.O. Box 4759, Santa Barbara, CA 93103,* has a catalog describing some 100 reports that give information on using a computer to make money. Write to them and ask for their catalog. Or, you might try phoning them at (805) 963-9580.

Rudy M. Hunger at **RMH Computers**, *P.O. Box 421, CH-6030 Ebikon, Switzerland,* says they have kits available that will act as an interface between the CE-158 RS-232 connector and a Brother EP-20. The EP-20 can then serve as a computer-driven printer.

The Radio Shack Model 100 portable continues to receive much praise. A college in Texas, **Dallas Baptist College**, has issued a requirement that *all* entering students (in all disciplines, not just science and engineering) be equipped with their own Model 100. Students pay for the unit through a special registration surcharge that is spread over four semesters. A college spokesman said in adopting the measure that: "Computer literacy is not an option for the educated man or woman of the 1980's. It's a requirement." A college spokesman also added: "We looked at eight or nine other computers that either fit in the pocket or in the briefcase. Radio Shack just simply did the best all-around job for the price."

Speaking of briefcase portables, it seems that the Sharp PC-5000 will not arrive in September. The latest word I have gleaned from unofficial sources is that it may be available by December. That should be just in time for it to land smack in the middle of a lot of other rumored entries to the field. The grapevine is now whispering that even outfits like Apple and Commodore are feverishly working on candidates for the "notebook" class.

I have heard that Sharp will soon be releasing a ROM programmer that will enable small businesses to "burn" their own PROMs. No solid information on capability and prices yet, but I have heard figures of $1500 for the system, $50 to $60 each for the PROMs. Figuring a markup of 2:1, that would mean software suppliers would have to be reselling the ROMs (with software installed) for $150 - $180 each. Whew! Can the market bear it? If you think so, ask a Sharp representative about it.

Sharp is also apparently on the verge of releasing a combination calculator/computer with a lot of transcendental functions available directly from the keyboard. I understand it will be called the model EL-5500. That indicates that it is considered part of their "calculator" (versus "computer") line. It seems to be similar to a 1250 with souped-up capabilities. The price will likely be $99.95 in the U.S. It seems there will also be a little printer made available for the EL-5500 that may sell as low as $70.00. Could it be that this little unit may also work with the PC-1250 in a pinch? It would sure save some money over the CE-125 if that turns out to be the case.

Say, did Sharp score one on Tandy? Radio Shack's PC-3 is a custom copy of the PC-1250. Now word is that Sharp will provide U.S. customers with a PC-1250A, which is known as a 1251 outside the states. This 4K version of the 1250 may be sold at about the same price as the Shack's PC-3. That should do wonders for their PC-3 sales.

By the way, note the $119.95 price on Radio Shack's Printer/Cassette Interface for the PC-3. Say, do you suppose you could just plug a 1250 into that and save $50 over the list price on the CE-125?

Next issue *PCN* will increase to 16 pages. This will necessitate our switching from first class to third class mailing. If you want to receive your copy of *PCN* in a timely manner, then please make sure that the address shown on the latest mailer is current. Third class mail is returned to the sender for remailing rather than automatically being forwarded by the postal service. Thus, it is vital that we have your correct address and that you keep us informed of any changes. Please forward any change of address notice to the attention of our Subscription Department.

P.O. Box 232, Seymour, CT 06483